

EXPERIMENTO 6 – DESLOCAMENTO LÓGICO E ARITMÉTICO

A. Introdução

O foco deste experimento é o estudo de operações de deslocamento dos tipos lógico e aritmético. Essas operações são muito utilizadas em sistemas digitais para, por exemplo, fazer a divisão e multiplicação de números por potências de 2. Para dar suporte ao experimento, faça o download, descompacte e abra no Quartus II o arquivo “Lab06_projeto.zip” disponível no site da disciplina. **Atenção: (i) Não se esqueça de utilizar a opção “Open Project...” para abrir o projeto; e (ii) se necessário, desmarque a opção Show Location Assignments do menu View (View → Show Location Assignments) para remover as caixas com rótulos iniciados por PIN que podem aparecer quando o projeto for aberto.**

B. Deslocamento lógico usando um barrel-shifter

O deslocamento lógico é o deslocamento realizado sobre os bits de um número binário acrescentando zeros às posições que se tornarem vazias e desprezando os bits que ficarem em posições não existentes em função do tamanho de palavra adotado. Para ilustrar as operações de deslocamento lógico, a Tabela 1 apresenta exemplos de tais operações.

Tabela 1. Demonstração de operações de deslocamento lógico à esquerda e à direita de números de 8 bits.

$10_{10} = $ <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td colspan="8">← 1x à esquerda:</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td colspan="8">← 4x à esquerda:</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td colspan="8">← 6x à esquerda:</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	1	0	1	0	← 1x à esquerda:								0	0	0	1	0	1	0	0	← 4x à esquerda:								1	0	1	0	0	0	0	0	← 6x à esquerda:								1	0	0	0	0	0	0	0	$142_{10} = $ <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td colspan="8">→ 1x à direita:</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td colspan="8">→ 3x à direita:</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td colspan="8">→ 6x à direita:</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table>	1	0	0	0	1	1	1	0	→ 1x à direita:								0	1	0	0	0	1	1	1	→ 3x à direita:								0	0	0	1	0	0	0	1	→ 6x à direita:								0	0	0	0	0	0	1	0
0	0	0	0	1	0	1	0																																																																																																										
← 1x à esquerda:																																																																																																																	
0	0	0	1	0	1	0	0																																																																																																										
← 4x à esquerda:																																																																																																																	
1	0	1	0	0	0	0	0																																																																																																										
← 6x à esquerda:																																																																																																																	
1	0	0	0	0	0	0	0																																																																																																										
1	0	0	0	1	1	1	0																																																																																																										
→ 1x à direita:																																																																																																																	
0	1	0	0	0	1	1	1																																																																																																										
→ 3x à direita:																																																																																																																	
0	0	0	1	0	0	0	1																																																																																																										
→ 6x à direita:																																																																																																																	
0	0	0	0	0	0	1	0																																																																																																										

O *barrel-shifter* é um circuito combinacional utilizado para fazer a operação de deslocamento lógico. Tal circuito pode ser obtido usando multiplexadores dispostos em diferentes estágios, cada um realizando uma quantidade de deslocamentos que é potência de dois. Através da ativação de cada um ou de ambos os estágios, é possível obter diferentes quantidades de deslocamentos. A Figura 1 apresenta a estrutura de um *barrel-shifter* que realiza o deslocamento de números binários de 4 bits para a direita. Faça a montagem de tal *barrel-shifter* seguindo os seguintes passos:

- 1) Crie um novo arquivo para esquemático denominado “**barrelshifter4**”;
- 2) Inclua as entradas e as saídas;
- 3) Faça a implementação usando o bloco **mux2x1** que já está disponível no projeto;
- 4) Crie o bloco “**barrelshifter4**” (utilizando a opção **File → Create / Update → Create Symbol Files for Current File**).

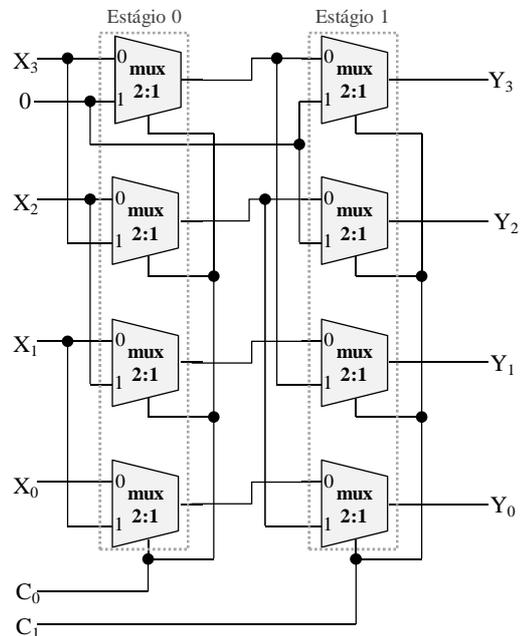


Figura 1. Estrutura de um barrel-shifter de 4 bits.

Com o *barrel-shifter* montado, inclua seu bloco na área de projeto e faça testes observando as seguintes características:

- 1) É possível realizar de 0 a 3 deslocamentos;
- 2) O deslocamento à direita resulta na divisão do número original por 2^N , onde N é o número de deslocamentos realizados. Para visualizar melhor essa característica, conecte as saídas do *barrel-shifter* também no bloco **bin4to7seg** disponível no projeto e observe os valores de saída em dois *displays* de 7 segmentos.

C. Deslocamento à esquerda

Avalie o funcionamento do circuito da Figura 1 e construa, a partir de tal circuito, um novo circuito/bloco “**barreshifter4esq**” que realize deslocamentos à esquerda.

D. Deslocamento aritmético

Conforme observado na seção anterior, a operação de deslocamento à direita implica dividir o número original por 2^N (N = número de deslocamentos). No entanto, se estivermos trabalhando com números representados em complemento de 2, o deslocamento lógico à direita não irá resultar em uma divisão por 2^N . Isso pode ser observado considerando, por exemplo, que a realização de dois deslocamentos à direita do número **-4** codificado usando complemento de 2 com 4 bits (**1100₂**) resulta no número **3** (**0011₂**). Para realizar divisões por 2^N também no caso de números em complemento de 2, é necessário realizar um deslocamento aritmético à direita, o qual está ilustrado na Tabela 2 (observe que o arredondamento da divisão é feito para o inteiro imediatamente abaixo do número real resultante da divisão). Comparando a Tabela 1 com a Tabela 2, é possível observar que enquanto no deslocamento lógico à direita são inseridos zeros nas posições vazias, no deslocamento aritmético o bit mais significativo do número original é o que é utilizado para preencher as posições que se tornaram vazias. Assim, no caso do deslocamento aritmético à direita, é necessário fazer uma **extensão do sinal**. No caso do deslocamento aritmético à esquerda não é realizada extensão do sinal e, portanto, esse deslocamento coincide com o deslocamento lógico à esquerda.

Tabela 2. Demonstração de operações de deslocamento aritmético de números de 8 bits.

$100_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table> $\rightarrow 1x \text{ à direita}$	0	1	1	0	0	1	0	0	$-32_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> $\rightarrow 1x \text{ à direita}$	1	1	1	0	0	0	0	0
0	1	1	0	0	1	0	0										
1	1	1	0	0	0	0	0										
$50_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table> $\rightarrow 3x \text{ à direita:}$	0	0	1	1	0	0	1	0	$-16_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> $\rightarrow 3x \text{ à direita:}$	1	1	1	1	0	0	0	0
0	0	1	1	0	0	1	0										
1	1	1	1	0	0	0	0										
$12_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table> $\rightarrow 6x \text{ à direita:}$	0	0	0	0	1	1	0	0	$-4_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table> $\rightarrow 6x \text{ à direita:}$	1	1	1	1	1	1	0	0
0	0	0	0	1	1	0	0										
1	1	1	1	1	1	0	0										
$1_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table>	0	0	0	0	0	0	0	1	$-1_{10} = $ <table border="1" style="display: inline-table; text-align: center;"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1										
1	1	1	1	1	1	1	1										

Conhecido o deslocamento aritmético, estude o funcionamento do *barrel-shifter* da Figura 1 e modifique o seu circuito para obter um outro *barrel-shifter* que realize operações de deslocamento aritmético à direita. Para melhor observar os resultados do deslocamento aritmético de números de 4 bits em complemento de 2, utilize o bloco “**compl2to7seg**” que está disponível no projeto para conectar a saída do seu *barrel-shifter* em displays de 7 segmentos.